

REMARKS

The Office Action mailed August 19, 2009, considered and rejected claims 1-4, 6-16 and 18. Claims 1-4, 6-16, and 18 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Baisley* et al., (U.S. Patent No. 6,106,574) in view of *Haikin* (U.S. Patent No. 6,757,893), *Murakami* (U.S. Publ. No. 2003/0167423) and *Boxall* et al. (U.S. (U.S. Patent No. 6,263,456).¹

By this paper, claims 1, 11 and 18 are amended, claims 19 and 20 are added, and no claims are cancelled. Accordingly, following entry of this paper, claims 1-4, 6-16 and 18-20 are pending, of which claims 1 and 11 are the independent claims at issue.

As reflected above, Applicants claims relate generally to associating original source code with binary code for debugging purposes. As reflected in claim 1, for example, a computer-readable storage medium has computer-executable instructions that are executable by a processor to implement a method for associating original source code with binary code for debugging the binary code. The method itself includes storing a source code file on a server, the source code file including source code and being associated with a version. The source code is compiled into a binary file and while being compiled, various types of information are extracted. That information includes: a location of the source code file, the version associated with the source code file, a name of the server, a port of the server at which the server may be accessed to access the source code, a path to the source code, and a numeric value that indicates a version number of the source code. The extracted information is stored in a debug file associated with the binary file, and the debug file is stored. The debug file is stored with information from a data stream divided into multiple sets of lines, including one or more lines of a global variables area, one or more lines of a local variables area, and one or more lines of a source files area. After compiling the source code file, an instruction for a debugger to debug the binary file is received. Thereafter, the extracted information in the debug file is used to locate the source code file and associated it with the binary file. The binary file is then debugged with full source code support by correlating lines of source code with binary instructions in the binary file. Claim 11 recites a similar system but with various additional elements not recited in claim 1.

Applicant respectfully submits that while *Baisley*, *Haikin*, *Murakami*, and *Boxall* generally relate to compiling source code and/or debugging code, they fail—whether cited alone or in combination—to disclose or reasonably support the pending claims as recited in their entireties. For

¹ Although the prior art status of the cited art is not being challenged at this time, Applicant reserves the right to challenge the prior art status of the cited art at any appropriate time, should it arise. Accordingly, any arguments and amendments made herein should not be construed as acquiescing to any prior art status of the cited art.

example, among other things, the cited art fails to disclose or reasonably support extracting debug information and storing it in a debug file that includes information read from a data stream that includes a global variables area, a local variables area, and a source files area as recited in the pending claims.

Baisley, for example, discloses a method for relating objects in a compiler to locations in a source program, and in which an object identifies a source location. In the system, an object includes only one instance variable, which is itself an integer. Notably, however, nothing appears to relate in any way to a data stream used to produce information stored in a debug file, let alone that the data stream includes three separate areas that each have their own lines, as recited in combination with the other claim elements. Accordingly, Applicant has not found, nor has the Office cited, any teaching or reasonable support within *Baisley* for a data stream with separate lines delineating global variables, local variables, and source files areas, as recited in combination with the other claim elements.

Applicant respectfully submits that *Haiken*, *Murakami*, and *Boxall* fail to remedy such deficiencies. For example, *Haiken* describes a version control system for software code, and includes representing initial versions of source lines that are stored at separate memory address locations. (Col. 10, ll. 58-62). However, Applicant has not found, nor has the Office cited, any teaching or reasonable support within *Haiken* for a data stream with separate lines delineating global variables, local variables, and source files areas, as recited in combination with the other claim elements.

With regard to *Murakami*, Applicant notes that *Murakami* relates to a system for testing consistency of code files and source files. In such a system, an XML file may be used to identify the location of an object. For example, a URL may be provided of the form: "http://sv1/mast/loc/obj/pg.o." Notably, however, the use of a URL clearly fails to disclose or reasonably support a global, local, and/or source files area as claimed in combination with the other art of record. In particular, such a URL merely identifies a location, and does not itself include a data stream, nor does it have separate sets of lines defining global variables, local variables, and/or source files areas. While such URL clearly fails to disclose or reasonably support the recited data stream, Applicant has not found any more relevant discussion in *Murakami*, nor has the Office cited, any teaching or reasonable support within *Murakami* for a data stream with separate lines delineating global variables, local variables, and source files areas, as recited in combination with the other claim elements.

Boxall provides no additional disclosure to remedy the deficiencies of *Baisley*, *Haiken* and *Murakami*. In particular, *Boxall* relates to a system for remote debugging of a client/server

application, and in which a _DBG_Start_parms) entry point is used to pass various parameters (e.g., network or IP address of client, PID of client, TID of client, network IP address of client where debugger is to run, port number of port where debugger listens). Such disclosure thus relates to passing IP and port information, but nothing in *Boxall* appears to teach or reasonably support a data stream with different lines defining a global variables, local variables, and/or source files areas as recited in combination with the other art of record.

In view of the foregoing, Applicant respectfully submits that the other rejections to the claims are now moot and do not, therefore, need to be addressed individually at this time. It will be appreciated, however, that this should not be construed as Applicant acquiescing to any of the purported teachings or assertions made in the last action regarding the cited art or the pending application, including any official notice. Instead, Applicant reserves the right to challenge any of the purported teachings or assertions made in the last action at any appropriate time in the future, should the need arise. Furthermore, to the extent that the Examiner has relied on any Official Notice, explicitly or implicitly, Applicant specifically requests that the Examiner provide references supporting the teachings officially noticed, as well as the required motivation or suggestion to combine the relied upon notice with the other art of record.

In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney at (801) 533-9800.

Dated this 19th day of November, 2009.

Respectfully submitted,

/Colby C. Nuttall, Reg. No. 58,146/
Colby C. Nuttall

RICK D. NYDEGGER
Registration No. 28,651
COLBY C. NUTTALL
Registration No. 58,146
Attorneys for Applicant
Customer No. 47973